Package 'isoorbi'

June 24, 2023

Type Package

Title Process Orbitrap Isotopocule Data

Version 1.1.0

Date 2023-6-22

URL https://github.com/isoverse/isoorbi

BugReports https://github.com/isoverse/isoorbi/issues

Depends R (>= 4.1.0)

Imports utils (>= 4.1.0), stats (>= 4.1.0), rlang (>= 1.0.0), lifecycle (>= 1.0.0), tidyr (>= 1.2.0), dplyr (>= 1.1.1), readr (>= 2.1.0), tidyselect (>= 1.2.0)

Suggests devtools, knitr, rmarkdown, testthat, scales, ggplot2, forcats

Description Read and process isotopocule data from an Orbitrap Isotope Solutions mass spectrometer. Hilkert et al. (2021) <doi:10.1021/acs.analchem.1c00944>.

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.2.3

NeedsCompilation no

Author Caj Neubauer [aut, cre, cph] (<https://orcid.org/0000-0002-5348-5609>), Sebastian Kopf [aut] (<https://orcid.org/0000-0002-2044-0201>), Kristýna Kantnerová [aut] (<https://orcid.org/0000-0001-6259-3225>)

Maintainer Caj Neubauer <caj.neubauer@colorado.edu>

Repository CRAN

Date/Publication 2023-06-24 18:20:06 UTC

R topics documented:

`orbi_adjust_block` *Manually adjust block delimiters*

Description

Note that adjusting blocks removes all block segmentation. Make sure to call `orbi_segment_blocks()` **after** adjusting block delimiters. FIXME: complete description and parameters

Usage

```
orbi_adjust_block(
  dataset,
  block,
  filename = NULL,
  shift_start_time.min = NULL,
  shift_end_time.min = NULL,
  shift_start_scan.no = NULL,
  shift_end_scan.no = NULL,
  set_start_time.min = NULL,
  set_end_time.min = NULL,
  set_start_scan.no = NULL,
  set_end_scan.no = NULL
)
```

Arguments

`dataset`	tibble produced by `orbi_define_blocks_for_dual_inlet()`
`block`	the block for which to adjust the start and/or end
`filename`	needs to be specified only if the `dataset` has more than one `filename`
`shift_start_time.min`	if provided, the start time of the block will be shifted by this many minutes (use negative numbers to shift back)
`shift_end_time.min`	if provided, the end time of the block will be shifted by this many minutes (use negative numbers to shift back)
`shift_start_scan.no`	if provided, the start of the block will be shifted by this many scans (use negative numbers to shift back)
`shift_end_scan.no`	if provided, the end of the block will be shifted by this many scans (use negative numbers to shift back)
`set_start_time.min`	if provided, sets the start time of the block as close as possible to this time
`set_end_time.min`	if provided, sets the end time of the block as close as possible to this time
`set_start_scan.no`	if provided, sets the start of the block to this scan number (scan must exist in the `dataset`)
`set_end_scan.no`	if provided, sets the end of the block to this scan number (scan must exist in the `dataset`)

Value

A data frame (tibble) with block limits altered according to the provided start/end change parameters. Any data that is no longer part of the original block will be marked with the value of `orbi_get_settings("data_type_unused")`. Any previously applied segmentation will be discarded (`segment` column set to `NA`) to avoid unintended side effects.

`orbi_calculate_ratio` *Calculate isotopocule ratio*

Description

This function calculates the ratio of two isotopocules (the `numerator` and `denominator`) by averaging multiple measurements of each using the `ratio_method` and returns a single value. Normally this function is not called directly by the user, but via the function `orbi_summarize_results()`, which calculates isotopocule ratios and other results for an entire dataset.

Usage

```
orbi_calculate_ratio(
  numerator,
  denominator,
  ratio_method = c("mean", "sum", "median", "geometric_mean", "slope", "weighted_sum")
)
```

Arguments

`numerator`	Column(s) used as numerator; contains ion counts
`denominator`	Column used as denominator; contains ion counts
`ratio_method`	Method for computing the ratio. **Please note well**: the formula used to calculate ion ratios matters! Do not simply use arithmetic mean. The best option may depend on the type of data you are processing (e.g., MS1 versus M+1 fragmentation). `ratio_method` can be one of the following: • `mean`: arithmetic mean of ratios from individual scans. • `sum`: sum of all ions of the numerator across all scans divided by the sum of all ions observed for the denominator across all scans. • `geometric_mean`: geometric mean of ratios from individual scans. • `slope`: The ratio is calculated using the slope obtained from a linear regression model that is weighted by the `numerator x`, using `stats::lm(x ~ y + 0, weights = x)`. • `weighted_sum`: A derivative of the `sum` option. The weighing function ensures that each scan contributes equal weight to the ratio calculation, i.e. scans with more ions in the Orbitrap do not contribute disproportionately to the total `sum` of `x` and `y` that is used to calculate `x/y`.

Value

Single value ratio between the isotopocules defined as `numerator` and `denominator` calculated using the `ratio_method`.

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <-
  orbi_read_isox(file = fpath) |>
  orbi_simplify_isox() |>
  orbi_define_basepeak(basepeak_def = "M0")

ratio <- orbi_calculate_ratio(
   numerator = df$ions.incremental,
   denominator = df$basepeak_ions,
   ratio_method = "sum")
```

orbi_calculate_ratios *Calculate isotopocule ratios (deprecated)*

Description

[Deprecated] This function was renamed to `orbi_calculate_ratio()` to better reflect what it does.

Usage

```
orbi_calculate_ratios(...)
```

Arguments

`...`	parameters passed on to new function orbi_calculate_ratio()

orbi_define_basepeak *Define and assign the denominator for ratio calculation*

Description

`orbi_define_basepeak()` sets one isotopocule in the data frame as the base peak (ratio denominator)

Usage

```
orbi_define_basepeak(dataset, basepeak_def)
```

Arguments

`dataset`	A tibble from a `IsoX` output. Needs to contain columns for `filename`, `compound`, `scan.no`, `isotopocule`, `ions.incremental`.
`basepeak_def`	The isotopocule that gets defined as base peak, i.e. the denominator to calculate ratios

Value

Input data frame without the rows of the basepeak isotopocule and instead two new columns called `basepeak` and `basepeak_ions` holding the basepeak information

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath) |>
  orbi_simplify_isox() |>
  orbi_define_basepeak(basepeak_def = "M0")
```

`orbi_define_blocks_for_dual_inlet` *Binning raw data into blocks for dual inlet analyses*

Description

Binning raw data into blocks for dual inlet analyses

Usage

```
orbi_define_blocks_for_dual_inlet(
  dataset,
  ref_block_time.min,
  change_over_time.min,
  sample_block_time.min = ref_block_time.min,
  startup_time.min = 0,
  ref_block_name = setting("di_ref_name"),
  sample_block_name = setting("di_sample_name")
)
```

Arguments

`dataset`	A data frame or tibble produced from IsoX data by `orbi_simplify_isox()`
`ref_block_time.min`	placeholder
`change_over_time.min`	placeholder
`sample_block_time.min`	placeholder
`startup_time.min`	placeholder
`ref_block_name`	placeholder
`sample_block_name`	placeholder

Value

A data frame (tibble) with block annotations in the form of the additional columns described below:

- `data_group` is an integer that numbers each data group (whether that's startup, a sample block, a segment, etc.) in each file sequentially to uniquely identify groups of data that belong together - this columns is NOT static (i.e. functions like `orbi_adjust_block()` and `orbi_segment_blocks()` will lead to renumbering) and should be used purely for grouping purposes in calculations and visualization
- `block` is an integer counting the data blocks in each file (0 is the startup block)

- sample_name is the name of the material being measured as defined by the ref_block_name and sample_block_name parameters
- segment is an integer defines segments within individual blocks - this will be NA until the optional orbi_segment_blocks() is called
- data_type is a text value describing the type of data in each data_group - for a list of the main categories, call orbi_get_settings("data_type")

---

orbi_filter_isox *Basic generic filter for IsoX data*

Description

A basic filter function orbi_filter_isox() for file names, isotopocules, compounds and time ranges. Default value for all parameters is FALSE, i.e. no filter is applied.

Usage

```
orbi_filter_isox(
  dataset,
  filenames = FALSE,
  compounds = FALSE,
  isotopocules = FALSE,
  time_min = FALSE,
  time_max = FALSE
)
```

Arguments

dataset	The IsoX data to be filtered
filenames	Vector of file names to keep
compounds	Vector of compounds to keep
isotopocules	Vector of isotopocules to keep
time_min	Minimum retention time in minutes (time.min)
time_max	Maximum retention time in minutes (time.min)

Value

Filtered tibble

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath) |>
orbi_simplify_isox() |>
orbi_filter_isox(filenames = c("s3744"),
compounds = "HSO4-",
isotopocules = c("M0", "34S", "18O"),
time_min = FALSE,
time_max = FALSE)
```

`orbi_filter_satellite_peaks`

Filter to remove minor satellite peaks

Description

Remove minor signals (e.g., satellite peaks) that were reported by IsoX

Usage

```
orbi_filter_satellite_peaks(dataset)
```

Arguments

`dataset`	A data frame or tibble produced from IsoX data by `orbi_simplify_isox()`

Details

The `orbi_filter_satellite_peaks()` function removes minor signals for an isotopocule that have been reported by IsoX. These are often small `satellite peaks` generated by the Fourier transform.

If there are signal of high intensity or very many signals, this can indicate that the m/z and tolerance setting used for processing .raw files with IsoX were incorrect.

Value

A filtered data frame (tibble)

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath) |>
orbi_simplify_isox() |>
orbi_filter_satellite_peaks()
```

orbi_filter_scan_intensity *Filter to remove extreme scans*

Description

The function orbi_filter_scan_intensity() removes extremely high and low intense scans based on TIC x injection time (i.e., ion intensity)

Usage

```
orbi_filter_scan_intensity(dataset, outlier_percent)
```

Arguments

dataset	Simplified IsoX dataset to have TICxIT outliers removed
outlier_percent	A number between 0 and 10. Remove this percentage of scans based on TIC multiplied by injection time.

Details

Function is intended to remove scans that are outliers. TIC multiplied by injection time serves as an estimate for the number of ions in the Orbitrap.

The filter is a basic truncation that removes x % of scans with the largest **and** x % of scans with the smallest ion estimates. Grouping is by columns filename and compound.

The input dataset is expected to have at least these 8 columns: filename, scan.no, time.min, compound, isotopocule, ions.incremental, tic, it.ms.

Value

Filtered tibble

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath) |>
orbi_simplify_isox() |>
orbi_filter_scan_intensity(outlier_percent = 1)
```

orbi_filter_weak_isotopocules *Filter to remove weak isotopocules*

Description

The function `orbi_filter_weak_isotopocules()` removes isotopocules that are not consistently detected in most scans

Usage

```
orbi_filter_weak_isotopocules(dataset, min_percent)
```

Arguments

`dataset`	A simplified IsoX data frame to be processed
`min_percent`	A number between 0 and 90. Isotopocule must be observed in at least this percentage of scans (please note: the percentage is defined relative to the most commonly observed isotopocule of each compound)

Details

The input `dataset` is expected to have at least these 8 columns: `filename`, `scan.no`, `time.min`, `compound`, `isotopocule`, `ions.incremental`, `tic`, `it.ms`.

Value

A filtered tibble (data frame)

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath) |>
      orbi_simplify_isox() |>
      orbi_filter_weak_isotopocules(min_percent = 2)
```

orbi_get_blocks_info *Summarize blocks info*

Description

FIXME: fully document

Usage

```
orbi_get_blocks_info(dataset)
```

Arguments

dataset	tibble produced by orbi_define_blocks_for_dual_inlet()

---

orbi_get_settings *Get all isoorbi package settings*

Description

Get all isoorbi package settings

Usage

```
orbi_get_settings(pattern = NULL)
```

Arguments

pattern	an optional parameter with a regular expression pattern by which to sub-select the returned settings

Value

list of all package settings and their values

Examples

```
orbi_get_settings()
```

---

orbi_read_isox *Read IsoX file*

Description

Read an IsoX output file (.isox) into a tibble data frame

Usage

```
orbi_read_isox(file)
```

Arguments

file	Path to the .isox file

Details

Additional information on the columns:

- `filename`: name of the original Thermo `.raw` file processed by IsoX
- `scan.no`: scan number
- `time.min`: acquisition or retention time in minutes
- `compound`: name of the compound (e.g., NO3-)
- `isotopocule`: name of the isotopocule (e.g., 15N); called `isotopolog` in `.isox`
- `ions.incremental`: estimated number of ions, in increments since it is a calculated number
- `tic`: total ion current (TIC) of the scan
- `it.ms`: scan injection time (IT) in millisecond (ms)

Value

A tibble containing at minimum the columns `filename`, `scan.no`, `time.min`, `compound`, `isotopocule`, `ions.incremental`, `tic`, `it.ms`

Examples

```
fpath <- system.file("extdata", "testfile_dual_inlet.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath)
```

---

orbi_segment_blocks *Segment data blocks*

Description

This step is optional and is intended to make it easy to explore the data within a sample or ref data block. Note that any raw data not identified with `data_type` set to "data" (`orbi_get_settings("data_type")`) will stay unsegmented. This includes raw data flagged as "startup", "changeover", and "unused".

Usage

```
orbi_segment_blocks(
  dataset,
  into_segments = NULL,
  by_scans = NULL,
  by_time_interval = NULL
)
```

Arguments

dataset	tibble produced by orbi_define_blocks_for_dual_inlet()
into_segments	segment each data block into this many segments. The result will have exactly this number of segments for each data block except for if there are more segments requested than observations in a group (in which case each observation will be one segment)
by_scans	segment each data block into segments spanning this number of scans. The result will be approximately the requested number of scans per segment, depending on what is the most sensible distribution of the data. For example, in a hypothetical data block with 31 scans, if by_scans = 10, this function will create 3 segments with 11, 10 and 10 scans each (most evenly distributed), instead of 4 segments with 10, 10, 10, 1 (less evenly distributed).
by_time_interval	segment each data block into segments spanning this time interval. The result will have the requested time interval for all segments except usually the last one which is almost always shorter than the requested interval.

---

orbi_set_settings *Set package settings*

Description

Use this function to change the default package settings. When calling this function, only specify the settings you want to change, everything else will remain unchanged. The default value for each parameter is what the package uses by default for each setting.

Usage

```
orbi_set_settings(
  di_ref_name = "ref",
  di_sample_name = "sam",
  data_type_data = "data",
  data_type_startup = "startup",
  data_type_changeover = "changeover",
  data_type_unused = "unused",
  reset_all = FALSE
)
```

Arguments

di_ref_name	the text label for dual inlet reference blocks
di_sample_name	the text label for dual inlet sample blocks
data_type_data	the text used to flag raw data as actually being data
data_type_startup	the text used to flag raw data as being part of the startup

`data_type_changeover`
: the text used to flag raw data as being part of a changeover

`data_type_unused`
: the text used to flag raw data as being unused

`reset_all`
: if set to TRUE, will reset all settings back to their defaults

Details

FIXME: needs documentation completion FIXME: needs tests to change settings and then get the value back

Value

invisible list of all settings (see `orbi_get_settings()`)

`orbi_simplify_isox` *Simplify IsoX data*

Description

Keep only columns that are directly relevant for isotopocule ratio analysis

Usage

```
orbi_simplify_isox(dataset)
```

Arguments

`dataset`
: IsoX data that is to be simplified

Value

A tibble containing only the 8 columns: `filename`, `scan.no`, `time.min`, `compound`, `isotopocule`, `ions.incremental`, `tic`, `it.ms`.

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package="isoorbi")
df <- orbi_read_isox(file = fpath) |> orbi_simplify_isox()
```

orbi_summarize_results *Generate the results table*

Description

Contains the logic to generate the results table. It passes the `ratio_method` parameter to the `orbi_calculate_ratio()` function for ratio calculations.

Usage

```
orbi_summarize_results(
  dataset,
 ratio_method = c("mean", "sum", "median", "geometric_mean", "slope", "weighted_sum"),
 .by = c("block", "sample_name", "segment", "data_group", "data_type", "injection")
)
```

Arguments

`dataset`	A tibble from `IsoX` output (`orbi_read_isox()`) and with a basepeak already defined (using `orbi_define_basepeak()`). Optionally, with block definitions (`orbi_define_blocks_for_dual_inlet()`) or even additional block segments (`orbi_segment_blocks()`).
`ratio_method`	Method for computing the ratio. **Please note well**: the formula used to calculate ion ratios matters! Do not simply use arithmetic mean. The best option may depend on the type of data you are processing (e.g., MS1 versus M+1 fragmentation). `ratio_method` can be one of the following: • `mean`: arithmetic mean of ratios from individual scans. • `sum`: sum of all ions of the numerator across all scans divided by the sum of all ions observed for the denominator across all scans. • `geometric_mean`: geometric mean of ratios from individual scans. • `slope`: The ratio is calculated using the slope obtained from a linear regression model that is weighted by the `numerator x`, using `stats::lm(x ~ y + 0, weights = x)`. • `weighted_sum`: A derivative of the `sum` option. The weighing function ensures that each scan contributes equal weight to the ratio calculation, i.e. scans with more ions in the Orbitrap do not contribute disproportionately to the total `sum` of `x` and `y` that is used to calculate `x/y`.
`.by`	additional grouping columns for the results summary (akin to dplyr's `.by` parameter e.g. in `dplyr::summarize()`). If not set by the user, all columns in the parameter's default values are used, if present in the dataset. Note that the order of these is also used to arrange the summary.

Value

Returns a results summary table retaining the columns `filename`, `compound`, `isotopocule` and `basepeak` as well as the grouping columns from the `.by` parameter that are part of the input `dataset`. Additionally this function adds the following results columns: `ratio`, `ratio_sem`, `ratio_relative_sem_permil`, `shot_noise_permil`, `No.of.Scans`, `minutes_to_1e6_ions`

- `ratio`: The isotope ratio between the `isotopocule` and the `basepeak`, calculated using the `ratio_method`
- `ratio_sem`: Standard error of the mean for the ratio
- `number_of_scans`: Number of scans used for the final ratio calculation
- `minutes_to_1e6_ions`: Time in minutes it would take to observe 1 million ions of the `isotopocule` used as numerator of the ratio calculation.
- `shot_noise_permil`: Estimate of the shot noise (more correctly thermal noise) of the reported ratio in permil.
- `ratio_relative_sem_permil`: Relative standard error of the reported ratio in permil

Examples

```
fpath <- system.file("extdata", "testfile_flow.isox", package = "isoorbi")
df <- orbi_read_isox(file = fpath) |>
      orbi_simplify_isox() |> orbi_define_basepeak(basepeak_def = "M0")  |>
      orbi_summarize_results(ratio_method = "sum")
```

Index